ML Techniques for Intrusion Detection in Cyber-Physical Systems

Subjects: Computer Science, Artificial Intelligence

Contributor: Vinícius F. Santos , Célio Albuquerque , Diego Passos , Silvio E. Quincozes , Daniel Mossé

Cyber-physical systems (CPS) are vital to key infrastructures such as Smart Grids and water treatment, and are increasingly vulnerable to a broad spectrum of evolving attacks. Whereas traditional security mechanisms, such as encryption and firewalls, are often inadequate for CPS architectures, the implementation of Intrusion Detection Systems (IDS) tailored for CPS has become an essential strategy for securing them. In this context, it is worth noting the difference between traditional offline Machine Learning (ML) techniques and understanding how they perform under different IDS applications.

cyber-physical systems | intrusion detection systems | offline machine learning

1. Introduction

Cyber-physical systems (CPSs) integrate sensing, processing, communication, actuation, and control through networks and physical devices [1][2]. CPSs are subject to diverse attacks that can affect different aspects of human life [3], particularly public infrastructures such as Smart Grids (SG), water treatment, and pipeline gas. CPSs are typically organized in three architectural layers: perception, transmission, and application [1]. The perception layer comprises sensor and actuator devices at the network's edge, exchanging data with the application layer through the intermediary transmission layer. Due to their resource constraints, perception layer devices are the most vulnerable to attacks [4][5].

As an example of CPS vulnerability, one of the most harmful attacks was performed by the Stuxnet malware that crippled Iran's Nuclear Program in 2010 [6]; it was designed to attack the Supervisory Control And Data Acquisition (SCADA) system used to control Iranian uranium enrichment centrifuges. Another example is the 2015 *blackEnergy* cyberattack that disrupted Ukraine's electricity service, impacting approximately 225,000 users.

The primary security mechanisms to protect CPSs devices from external attacks rely on encryption, firewalls, and antivirus systems. However, these mechanisms cannot avoid all attacks, especially considering that attackers are constantly evolving their strategies. In this context, using Intrusion Detection Systems (IDS) is fundamental for detecting malicious behavior and defending the CPSs from threats. IDSs may employ Machine Learning (ML) techniques to detect malicious activities by relying on training datasets [7][8][9]. However, many studies in the literature still use datasets collected from general internet protocols [5][10][11][12][13]. These datasets are not suitable for intrusion detection in CPSs, as they have little relationship with the actual existing equipment and lack traffic from typical CPS protocols.

The traditional offline ML techniques do not have their models frequently updated when behavior shifts occur. Today, with the increasing emergence of Big Data systems producing a huge volume of heterogeneous data and the tendency to take data processing to the network nodes [14], there is a need to classify attacks in real time from a large flow of data without compromising the hardware resources, such as memory and CPU. Therefore, traditional offline ML techniques may not be suitable for processing events from large data streams. On the other hand, the state-of-art techniques that support online learning processing (online ML) are still not extensively studied.

2. Offline and Online ML

Offline and online ML have different main characteristics, which offer advantages depending on the application. In offline ML, the model is trained and tested on a fixed dataset, which must be available ahead of time. Additionally, offline ML typically requires longer training time as it involves processing the entire dataset ahead of time, which can be resource-intensive, often requiring substantial computational resources to train and test the model on large fixed datasets.

One of the disadvantages of offline ML is the lack of ability to detect *Concept Drift*, a statistical distribution change in the variables or input attributes over time. In other words, the relationship between the attributes and the classes may evolve, making the model trained on old data less accurate or even invalid for making predictions on new data.

On the other hand, online ML creates the model incrementally, updating the model as new data arrive. It operates with immediate responses able to adapt to evolving patterns or changes in the process [15] with the use of change detectors. The model is designed to provide predictions or actions based on the incoming data, facilitating quick decision-making. So, the online ML technique is used to continuously monitor sensor data, identify anomalies, predict failures, and make real-time adjustments to optimize the CPS process. Additionally, it can be more resource-efficient as it learns and trains from streaming data indefinitely without compromising memory or process [16].

An online ML architecture is given in **Figure 1**: the classification process, which is divided into two stages [15], the loss function, and change detector with the warning or drift flag:

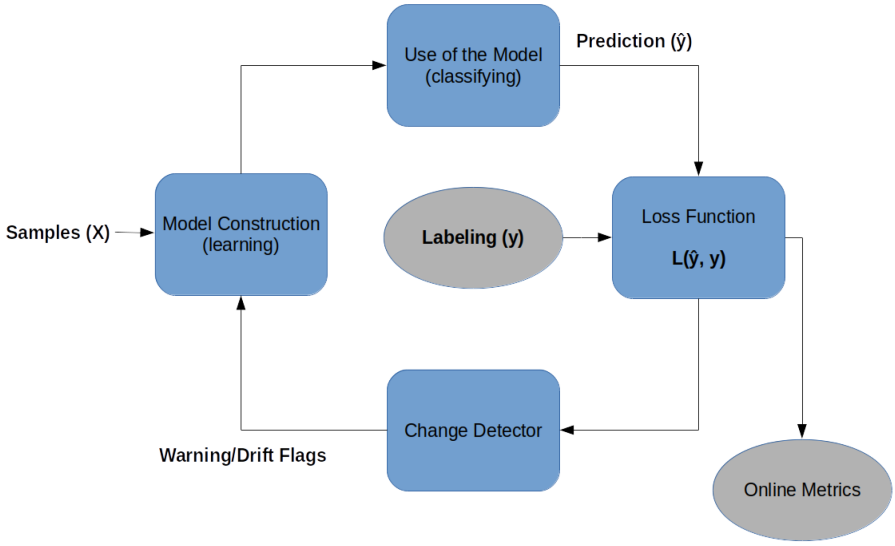

Figure 1. Outline of the online ML with prediction loss function and change detector.

- **Online model construction (learning)**: each new sample is used to build the model to determine the class with the attributes and values. This model can be represented by classification rules, decision trees, or mathematical formulas. To control the amount of memory used by the model, a *forgetting mechanism* is used, like a sliding window or fading factor, that discards samples while keeping an aggregate summary/value of these older samples;

- **Use of the model (classifying)**: it is the classification or estimation of unknown samples using the constructed model. In this step, the model's performance metrics are also calculated, comparing the known label of the sample *y* with the classified result of the model (i.e., the predicted label (y^)), for a supervised classification process.

- **Loss function**: for each input sample attributes (X), the prediction loss can be estimated as $L(\hat{y},y)$. The performance metrics of this method are obtained from the cumulative sum of sequential loss over time, that is, the loss function between forecasts and observed values.

- **Change detector**: detects concept drift by monitoring the loss estimation. When change is detected based on a predefined threshold, the warning signal or drift signal is used to retrain the model [17]. Interestingly, as will be shown below, no concept drift was detected in any of the datasets used in this article.

The evaluation of classical learning methods that use finite sets of data, such as cross-validation and training/test techniques, are not appropriate for data streams due to the unrestricted dataset size and the non-stationary independent samples. Two alternatives are more usual for data stream [18]:

- **Holdout**: applies tests and training samples to the classification model at regular time intervals (configurable by the user);

- **Prequential**: all samples are tested (prediction) and then used for training (learning).

The most suitable methodology for online ML models dealing with concept drift is the prequential method [18][19]. It combines the predictive and sequential aspects with memory window mechanisms, which maintain an aggregated summary of recent samples while discarding older ones in order to process new samples. This approach is based on the notion that statistical inference aims to make sequential probability predictions for future observations, rather than conveying information about past observations.

In online supervised ML the data-predicting process can be conducted in real time through the forecast model (**Figure 1**) to identify anomalies or failures in an industrial process. The labeling process is provided from CPS sensors after an additional delay. For example, consider a smart grid system where sensors continuously monitor power generation and distribution. In the absence of labeled data in real time, data prediction is crucial to detect anomalies or failures in the grid, thereby alerting operators to potential issues. By employing the labeling from the sensors after a delay, the system can analyze the loss and make corrections to the prediction model. The datasets presented in this article already include their respective classification labels, thereby enabling the use of supervised approaches for both online and offline algorithms.

The following online data stream classifiers will be used in this article because they were the most used in the studied references [20][21]:

- Naive Bayes (NB) is a probabilistic classifier, also called simple Bayes classifier or independent Bayes classifier, capable of predicting and diagnosing problems through noise-robust probability assumptions;
- Hoeffding Tree (HT) combines the data into a tree while the model is built (learning) incrementally. Classification can occur at any time;
- Hoeffding Adaptive Tree (HAT) adds two elements to the HT algorithm: change detector and loss estimator, yielding a new method capable of dealing with concept change. The main advantage of this method is that it does not need to know the speed of change in the data stream.

For the offline ML, the researchers selected five popular classifiers from the literature [16]:

- Naive Bayes (NB) is available for both online and offline ML versions, making it possible to compare them;
- Random Tree (RaT) builds a tree consisting of K randomly chosen attributes at each node;
- J48 builds a decision tree by recursively partitioning the data based on attribute values, it quantifies the randomness of the class distribution within a node (also known as entropy), to define the nodes of the tree;
- REPTree (ReT) consists of a fast decision tree learning algorithm that, similarly to J48, is also based on entropy;

- Random Forest (RF) combines the prediction of several independent decision trees to perform classification. Each tree is constructed to process diverse training datasets, multiple subsets of the original training data. The trees are built by randomly chosen attributes used to divide the data based on the reduction in entropy. The results of all trees are aggregated to decide on the final class based on majority or average, across all trees.

Four traditional ML metrics (overall accuracy (Acc); precision (P); recall (R); and F1-score) [1] will be used in this article to evaluate the performance of the classification based on the numbers of true positive (TP), true negative (TN), false positive (FP), and false negative (FN). Those metrics are defined as follows [16][22]:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

$$F1Score = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

3. CPS Related Datasets

Several datasets are widely used in IDS research, such as KDD Cup [10], CICIDS [12], UNSW-NB15 [13], and SUTD-IoT [23]. They are often cited in the literature, but these datasets are only based on classic IT networks and, as such, they are not representative of the traffic in industrial facilities networks. Therefore, they are not suitable for building or evaluating an IDS that operates with CPSs.

Seven specific CPS datasets can be seen in **Table 1** and the percentage of samples that represent attacks, known as *balance*. The Secure Water Treatment (SWaT) [3] and BATtle of the Attack Detection ALgorithms (BATADAL) [24] datasets are related to large-scale water treatment. The Morris datasets are related to Industrial Control System (ICS) and they are analyzed by Hink et al. [25], Morris and Gao [26], and Morris et al. [27]. ERENO is a Smart Grid (SG) dataset [22].

Table 1. CPS datasets profile.

Dataset	Attack Types	Number of Samples	Attributes	Balance (% Attack Class)
SWaT	1	946,719	51	5.77%
BATADAL	14	13,938	43	1.69%
Morris-1	40	76,035	128	77.85%
Morris-3 gas	7	97,019	26	36.96%
Morris-3 water	7	236,179	23	27.00%
Morris-4	35	274,628	16	21.87%
Ereno	7	5,750,000	69	6.60%

The SWaT dataset [3] was created for specific studies in the CPS area. It is available from the iTrust Lab website and was generated from an actual CPS for a modern six-stage water treatment system. In total, it contains 946,722 samples, each with 53 attributes collected from sensors and actuators in 11 days. The attack models of this dataset comprise several variations of fake data injection. In summary, the attacker captures the network messages to manipulate their values. Then, the tampered packets are transmitted to the PLCs. All attacks succeeded. All the attacks on SWaT dataset appear during the last 4 days of data collection—which corresponds to approximately 1313 of the samples. As can be seen in Equations (2)–(4), precision, recall, and F1-score can only be defined in the presence of attacks—so that TP, FP, and FN are not all zero. Thus, for SWaT, those metrics can only be computed for the final third of the dataset.

Another dataset, known by the academic competition BATADAL [24], was created to test attack detection algorithms in CPS water distribution systems. The dataset consisted of a network of 429 pipes, 388 junctions, 7 tanks, 11 hydraulic pumps, 5 valves, and 1 reservoir. In total, there are nine PLCs to control the status of the valves (open or closed), the inlet pressure, and the outlet pressure of the pumps, as well as the flow of water. It contains 13,938 samples, 43 attributes, and 14 attack types, and it is available for the academic community. The attacks on BATADAL dataset only appear during the last 1515 of the dataset, when their occurrences must be identified in the competition. Therefore, like with the SWaT dataset, precision, recall, and F1-score can only be defined after the occurrence of these attacks.

Morris-1 [25] consists of a set of supervisory controls interacting with various SG devices along with network monitoring devices. The network consists of four circuit breakers controlled by smart relays, which are connected to a switch substation, through a router, to the supervisory control and the network acquisition system. The attack scenarios were built on the assumption that the attacker already has access to the substation network and can inject commands from the switch. In total, 76,035 samples were collected containing 128 attributes and four attacks.

The datasets on Morris-3 [26] were captured using network data logs that monitor and store MODBUS traffic from an RS-232 connection. Two laboratory-scale SCADA systems of a pipeline network (dataset Morris-3 gas) and a water storage tank (dataset Morris-3 water) were used. The Wireshark program was used to capture and store network traffic during normal and under-attack operations. The datasets have 97,019 samples for the gas system (Morris-3 gas) and 236,179 samples for the water system (Morris-3 water), with 26 and 23 attributes, respectively, and seven attacks each.

The Morris-4 dataset [27] also refers to a pipeline network simulation. This dataset has the same origin as the Morris-3 gas, but it has improvements such as 35 labeled random cyberattacks simulated in a virtual gas network. The virtual environment was chosen because it allows other experiments without the need to have access to physical devices and the possibility of expansion. The dataset has 274,628 samples with 16 attributes.

ERENO [22] was developed as a synthetic traffic generation framework based on the IEC-61850 [28] standard for SG. Its development was motivated by the absence of specific security datasets for SG. Therefore, the objective is to provide a reference for research in the area of intrusion detection. The dataset starts with the SG generation by means of simulations using the Power Systems Computer-Aided Design (PSCAD) tool. This allows the generation of realistic data from the Generic Object Oriented Substation Events (GOOSE) and Sampled Value (SV) protocols. In the end, the final dataset is composed of samples from seven different Use Cases (scenarios) in sequence (i.e., each at a different time), corresponding to seven different new sequential attacks, as well as normal operations.

4. Main Results

By evaluating various online and offline machine learning techniques for intrusion detection in the CPS domain, this study reveals that: (i) when known signatures are available, offline techniques present higher precision, recall, accuracy, and F1-Score, whereas (ii) for real-time detection, online methods are more suitable, as they learn and test almost 10 times faster. Furthermore, while offline ML's robust classification metrics are important, the adaptability and real-time responsiveness of online classifiers cannot be overlooked. Thus, a combined approach may be key to effective intrusion detection in CPS scenarios.

References

1. Quincozes, S.E.; Passos, D.; Albuquerque, C.; Ochi, L.S.; Mossé, D. GRASP-Based Feature Selection for Intrusion Detection in CPS Perception Layer. In Proceedings of the 2020 4th Conference on Cloud and Internet of Things (CIoT), Niteroi, Brazil, 7–9 October 2020; pp. 41–48.
2. Reis, L.H.A.; Murillo Piedrahita, A.; Rueda, S.; Fernandes, N.C.; Medeiros, D.S.; de Amorim, M.D.; Mattos, D.M. Unsupervised and incremental learning orchestration for cyber-physical security. Trans. Emerg. Telecommun. Technol. 2020, 31, e4011.

3. Goh, J.; Adepu, S.; Junejo, K.N.; Mathur, A. A Dataset to Support Research in the Design of Secure Water Treatment Systems. In Proceedings of the Critical Information Infrastructures Security, 11th International Conference, CRITIS 2016, Paris, France, 10–12 October 2016; Springer: Cham, Switzerland, 2017; pp. 88–99.
4. Obert, J.; Cordeiro, P.; Johnson, J.T.; Lum, G.; Tansy, T.; Pala, N.; Ih, R. Recommendations for Trust and Encryption in DER Interoperability Standards; Technical Report; Sandia National Lab (SNL-NM): Albuquerque, NM, USA, 2019.
5. Almomani, I.; Al-Kasasbeh, B.; Al-Akhras, M. WSN-DS: A dataset for intrusion detection systems in wireless sensor networks. J. Sensors 2016, 2016, 4731953.
6. Langner, R. Stuxnet: Dissecting a cyberwarfare weapon. IEEE Secur. Priv. 2011, 9, 49–51.
7. Kim, S.; Park, K.J. A Survey on Machine-Learning Based Security Design for Cyber-Physical Systems. Appl. Sci. 2021, 11, 5458.
8. Rai, R.; Sahu, C.K. Driven by Data or Derived Through Physics? A Review of Hybrid Physics Guided Machine Learning Techniques with Cyber-Physical System (CPS) Focus. IEEE Access 2020, 8, 71050–71073.
9. Mohammadi Rouzbahani, H.; Karimipour, H.; Rahimnejad, A.; Dehghantanha, A.; Srivastava, G. Anomaly Detection in Cyber-Physical Systems Using Machine Learning. In Handbook of Big Data Privacy; Springer International Publishing: Cham, Switzerland, 2020; pp. 219–235.
10. Lippmann, R.P.; Fried, D.J.; Graf, I.; Haines, J.W.; Kendall, K.R.; McClung, D.; Weber, D.; Webster, S.E.; Wyschogrod, D.; Cunningham, R.K.; et al. Evaluating Intrusion Detection Systems: The 1998 DARPA Off-Line Intrusion Detection Evaluation. In Proceedings of the DARPA Information Survivability Conference and Exposition, Hilton Head, SC, USA, 25–27 January 2000; Volume 2, pp. 12–26.
11. Tavallaee, M.; Bagheri, E.; Lu, W.; Ghorbani, A.A. A Detailed Analysis of the KDD CUP 99 Data Set. In Proceedings of the 2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications, Ottawa, ON, Canada, 8–10 July 2009; pp. 1–6.
12. Sharafaldin, I.; Lashkari, A.H.; Ghorbani, A.A. Toward generating a new intrusion detection dataset and intrusion traffic characterization. ICISSp 2018, 1, 108–116.
13. Moustafa, N.; Slay, J. UNSW-NB15: A Comprehensive Data Set for Network Intrusion Detection Systems (UNSW-NB15 Network Data Set). In Proceedings of the 2015 Military Communications and Information Systems Conference (MilCIS), Canberra, ACT, Australia, 10–12 November 2015; pp. 1–6.
14. Kartakis, S.; McCann, J.A. Real-Time Edge Analytics for Cyber Physical Systems Using Compression Rates. In Proceedings of the 11th International Conference on Autonomic Computing (ICAC 14), Philadelphia, PA, USA, 23–23 July 2014; pp. 153–159.

15. Hidalgo, J.I.G.; Maciel, B.I.; Barros, R.S. Experimenting with prequential variations for data stream learning evaluation. Comput. Intell. 2019, 35, 670–692.
16. Witten, I.H.; Frank, E. Data mining: Practical machine learning tools and techniques with Java implementations. ACM Sigmod. Rec. 2002, 31, 76–77.
17. Nixon, C.; Sedky, M.; Hassan, M. Practical Application of Machine Learning Based Online Intrusion Detection to Internet of Things Networks. In Proceedings of the 2019 IEEE Global Conference on Internet of Things (GCIoT), Dubai, United Arab Emirates, 4–7 December 2019; pp. 1–5.
18. Gama, J.; Sebastiao, R.; Rodrigues, P.P. Issues in Evaluation of Stream Learning Algorithms. In Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Paris, France, 28 June–1 July 2009; pp. 329–338.
19. Bifet, A.; Holmes, G.; Pfahringer, B.; Kranen, P.; Kremer, H.; Jansen, T.; Seidl, T. Moa: Massive Online Analysis—A Framework for Stream Classification and Clustering. In Proceedings of the First Workshop on Applications of Pattern Analysis, Windsor, UK, 1–3 September 2010; pp. 44–50.
20. Adhikari, U.; Morris, T.H.; Pan, S. Applying hoeffding adaptive trees for real-time cyber-power event and intrusion classification. IEEE Trans. Smart Grid 2017, 9, 4049–4060.
21. Domingos, P.; Hulten, G. Mining High-Speed Data Streams. In Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Boston, MA, USA, 20–23 August 2000; pp. 71–80.
22. Quincozes, S.E.; Albuquerque, C.; Passos, D.; Mossé, D. ERENO: An Extensible Tool For Generating Realistic IEC-61850 Intrusion Detection Datasets. In Proceedings of the Anais Estendidos do XXII Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais, Santa Maria, Brazil, 12–15 September 2022; pp. 1–8.
23. Aung, Y.L.; Tiang, H.H.; Wijaya, H.; Ochoa, M.; Zhou, J. Scalable VPN-Forwarded Honeypots: Dataset and Threat Intelligence Insights. In Proceedings of the Sixth Annual Industrial Control System Security (ICSS), Austin, TX, USA, 8 December 2020; pp. 21–30.
24. Taormina, R.; Galelli, S.; Tippenhauer, N.O.; Salomons, E.; Ostfeld, A.; Eliades, D.G.; Aghashahi, M.; Sundararajan, R.; Pourahmadi, M.; Banks, M.K.; et al. Battle of the attack detection algorithms: Disclosing cyber attacks on water distribution networks. J. Water Resour. Plan. Manag. 2018, 144, 04018048.
25. Hink, R.C.B.; Beaver, J.M.; Buckner, M.A.; Morris, T.; Adhikari, U.; Pan, S. Machine Learning for Power System Disturbance and Cyber-Attack Discrimination. In Proceedings of the 2014 7th International symposium on resilient control systems (ISRCS), Denver, CO, USA, 19–21 August 2014; pp. 1–8.

26. Morris, T.; Gao, W. Industrial Control System Traffic Data Sets for Intrusion Detection Research. In Proceedings of the Critical Infrastructure Protection VIII, 8th IFIP WG 11.10 International Conference (ICCIP 2014), Arlington, VA, USA, 17–19 March 2014; Revised Selected Papers 8. Springer: Cham, Switzerland, 2014; pp. 65–78.
27. Morris, T.H.; Thornton, Z.; Turnipseed, I. Industrial Control System Simulation and Data Logging for Intrusion Detection System Research. In Proceedings of the 7th Annual Southeastern Cyber Security Summit, Huntsville, AL, USA, 3–4 June 2015; pp. 3–4.
28. IEC-61850; Communication Networks and Systems in Substations. International Electrotechnical Commission: Geneva, Switzerland, 2003.